



Article

Improving Effectiveness of a Coaching System through Preference Learning [†]

Martin Žnidaršič * , Aljaž Osojnik , Peter Rupnik and Bernard Ženko *

Jožef Stefan Institute, Jamova Cesta 39, SI-1000 Ljubljana, Slovenia; aljaz.osojnik@ijs.si (A.O.); peter.rupnik@ijs.si (P.R.)

* Correspondence: martin.znidarsic@ijs.si (M.Ž.); bernard.zenko@ijs.si (B.Ž.)

[†] This paper is an extended version of our paper published in PETRA 2021: The 14th PErvasive Technologies Related to Assistive Environments Conference, Corfu, Greece, 29 June–2 July 2021.

Abstract: The paper describes an approach for indirect data-based assessment and use of user preferences in an unobtrusive sensor-based coaching system with the aim of improving coaching effectiveness. The preference assessments are used to adapt the reasoning components of the coaching system in a way to better align with the preferences of its users. User preferences are learned based on data that describe user feedback as reported for different coaching messages that were received by the users. The preferences are not learned directly, but are assessed through a proxy—classifications or probabilities of positive feedback as assigned by a predictive machine learned model of user feedback. The motivation and aim of such an indirect approach is to allow for preference estimation without burdening the users with interactive preference elicitation processes. A brief description of the coaching setting is provided in the paper, before the approach for preference assessment is described and illustrated on a real-world example obtained during the testing of the coaching system with elderly users.

Keywords: machine learning; preference learning; user centered design



Citation: Žnidaršič, M.; Osojnik, A.; Rupnik, P.; Ženko, B. Improving Effectiveness of a Coaching System through Preference Learning.

Technologies **2022**, *10*, 24.

<https://doi.org/10.3390/technologies10010024>

Academic Editor: Pedro Antonio Gutiérrez

Received: 31 December 2021

Accepted: 21 January 2022

Published: 31 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent decades, demographic and societal changes are causing ever more people to live alone at an older age. The desire to live independently, coupled with the unavailability of care facilities and their associated costs, is fueling interest in ambient assisted living (AAL) and sensor based monitoring and coaching solutions in one's home that would prolong independent living, ensure the safety and improve the quality of life of the elderly.

The SAAM coaching system, which was developed in scope of the collaborative research project of the same name (Supporting Active Aging through Multimodal coaching, <https://saam2020.eu/>, accessed on 14 January 2021) is one such solution. Although it features an array of sensor-based monitoring technologies, a situation awareness reasoning engine and automatically triggered coaching actions, as is common to many other systems of this kind, the SAAM coaching system additionally involves people from the user's social circles in the coaching loop. Namely, the system's outputs are meant not only for the supported user, but, situationally, also members of the user's social circle. For example, when the system suggests that some physical activity is advisable to maintain the user's usual level of activity, instead of messaging the user directly, the system may opt to send a message to the user's friend or relative to encourage them to invite the user for a joint walk.

Besides involving the user's social circle in coaching, the system has a friendly user interface, attempts to unobtrusively (as possible) sense and actuate coaching, and, crucially for this paper, attempts to adapt some of the parameters of the coaching process according to the user's preferences. To this end, the system employs a preference learning approach in order to infer the user preferences, which in turn affects how the preference-dependent components of system operate.

Namely, even though the reasoning models select suitable coaching actions based on the sensor-assessed context, in certain situations multiple coaching actions are suitable; this leaves room for adaptation to individual users or groups of users. Additionally, there are optional or even arbitrarily chosen parameter values related to rendering, i.e., the method of delivery and presentation of the coaching actions, that can be adjusted to suit the users and their preferences.

The main contribution of this paper, which is an extended version of the paper presented at the PETRA 2021 conference [1], is presentation of a novel approach for the assessment of user's preferences, its implementation and some illustrative examples of its use. While this solution is specifically tailored to the SAAM coaching system, the general approach could also be used with other similar systems.

The contents of the remainder of the paper are as follows. In Section 2 we present a selection of most relevant related work, and in Section 3 we briefly describe the specific coaching system, which of its parameters can be adapted from the learned preferences, and how that is achieved. Section 4 describes the main contribution of the paper, the preference learning approach, which is augmented by real-world illustrative examples. We conclude with a discussion in Section 5 and some final remarks in Section 6.

2. Related Work

Preferences are a concept present in many fields of science, ranging from economic decision theory [2] to AI [3]. There were also specific efforts [4] to provide an overview of preference modelling from aspects of various disciplines.

Preferences are especially important in multi-attribute decision theory [5,6], where they are commonly modelled with preference relations which are suited for modelling relative preferences or with value functions that can describe absolute preferences as aggregates of their sub-components which finally consider value functions of individual attributes of a given alternative (also denoted as marginal value functions). These are the most similar to the concepts we are aiming to learn.

Given an attribute, X , and one of its possible values, x , a value function $v(x)$ models the value of x for a decision maker and allows the modeling of relations to other possible attribute values, such as whether they have equal value to the user or whether some are preferred over the others and by how much. Value functions are commonly scaled to the interval $[0, 1]$. Some examples of possible value functions are provided in Figure 1.

The attributes that we are interested in (coaching action, persuasion strategy, and interaction target) are nominal, such as example (c) in Figure 1. In this nominal case, the value function for such an attribute forms a discrete distribution. Learning such a distribution in a given context is the goal of our approach, as we can use it in a distribution-based sampling procedure which ensures that the outputs of the coaching system are aligned with the user preferences.

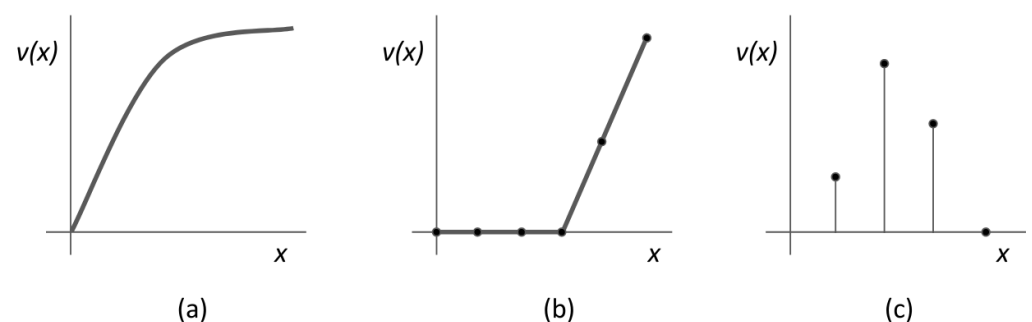


Figure 1. Examples of value functions: (a) a continuous value function that might hold, for example, for the value of one's salary; (b) value function for an ordinal attribute, for example, for the value of car's safety measured with NCAP points (in the example shown the user would be indifferent regarding values lower than 3); and (c) value function in case of a nominal attribute, such as color.

A value function can be estimated in a number of ways. In decision analysis, they are usually elicited by having experts either directly or indirectly interact with the parameters that affect the value functions. Some approaches of this kind are described and analyzed in [7,8]. These methods, however, are mainly based on observing decisions of actual people, and as such can induce unintended biases into the estimated value function. PRIME [9] extends these approaches to work with incomplete data and improves the estimation of value functions based on several theoretical insights. The interactive value function assessment approaches rely on interviews with experts or observation of their decisions. Acquiring more data in these kinds of scenarios requires additional expertise and effort to codify the preferences. This kind of procedures are likely to be costly and time consuming. On the other hand, in case of approaches based on (automatically) recorded data and machine learning, the biases will result from those that are present in the data and those stemming from the configuration of the machine learning procedure. Avoiding the former would be a significant effort in this context, while the biases due to the use of machine learning will be consistent across any functions that are estimated as long as the procedure is static.

Similar to ours, there are also primarily data-driven approaches to value function estimation, such as the UTA family of methods [10] that can be used to learn piecewise linear value functions from ranked alternatives with linear programming. These methods inspired numerous methodological extensions, from new approaches for learning with uncertainty [11] to interactive learning improvements [12] and AI assisted learning [13].

There are also evolutionary approaches aimed at learning value functions [14] or preference relations [15]. These approaches are generally interactive in that a user is able to directly integrate some of their preferences by providing pairwise comparisons. Thus, the evolutionary process that generates candidates for the value function estimation is guided according to the user's preferences.

The evolutionary approaches look at the problem of value function estimation as an (multi-objective) optimization problem. They differ from machine-learning based approaches in that they first elicit some decision makers preferences and based on those generate multiple candidates for the estimations which are then subjected to an evolutionary-genetic loop consisting of mating, offspring generation and selection. When a stopping criterion is met, the loop stops and the user is presented with a population (set) of candidates for the value function. Conversely, our approach infers preference value estimations based on (indirect or inferred) records of the context and outcome of the users interaction with the preference scenario by using a machine learning approach.

Of course, our approach is not the only one to utilize machine learning for learning about preferences. In general, preference learning can be also explored as a machine learning discipline [3,16]. In particular, the breadth of the field produces a plethora of approaches and application areas, from looking at how recommender systems can utilize preference learning [17], to predicting a choice (preference) from among a set of products [18] using particular machine learning approaches. For example, the INFINGER [19] is an interactive approach that addresses the same optimization problem as the evolutionary methods mentioned above, but does so with the use of a machine learning based inference mechanism in place of the evolutionary loop. On the other hand, ref. [20] utilizes an evolutionary loop on machine learning methods, in particular artificial neural networks.

Our approach is distinct in the sense that we aim for explicit representations of the value functions of selected attributes (as we need them for random sampling), which is characteristic for some decision modelling approaches, but instead of interactive elicitation that is commonly used in that field, we use a data-based approach for value function estimation based on indirect information. In particular, we are trying to estimate a user's preference of the delivery of the coaching actions in specific contexts, based on how they have responded to other coaching actions and the associated contexts in the past, be it positive (action completed) or negative (action declined or not possible).

In our case, we do not have direct access to the experts, i.e., primary users, to perform interactive preference assessment, such as those utilized in some of the approaches mentioned earlier in this section. As we only have available indirect indications of preference, we use data-based modeling to approximate the value function of each of our attributes, i.e., context factors, to determine what is their impact on the coaching action outcome. Notably, such an adaptive approach suffers from the “cold start problem”, where it can only be used once enough data have accumulated about user’s interaction with the system.

While in our illustrative examples in Section 4.3 we use two particular machine learning methods, decision trees and random forest, in principle, any arbitrary machine learning method can be used in their place.

3. Adapting Coaching to User’s Preferences

3.1. The SAAM Coaching System

The SAAM coaching system is a coaching system mainly targeted at the elderly that aims to support its users to continue living independently and comfortably for as long as possible. In contrast to typical AAL coaching approaches [21], it does not exclusively rely on directly delivering the coaching messages to users, but primarily focuses on delivering it to their social circles: family, friends, neighbors, and organized caregivers.

In the context of our system, we differentiate two types of users: *primary users (PUs)* are the ones that the system is providing coaching for, while the *secondary users (SUs)* are members of the primary users’ social circles. The operation of the SAAM coaching system using this terminology can then be stated as: the system seeks to assist primary users in living as independently as possible for as long as possible by predominantly directing the coaching messages to their associated *secondary users*, i.e., members of their social circles. The secondary users are then expected to act on the coaching message by interacting with the primary user in a way consistent with the message.

As an example, the system might detect that lately the primary user is not sleeping well, and issues the following coaching message to one or more of their associated SUs: “*Instruct your PU to only go to bed if they are sleepy*”. The SU can then pay a visit to or call the PU, ask about the quality of their sleep, and suggest to only go to bed if they are sleepy and not to engage in other activities, such as reading or watching TV, a factor that negatively impacts sleep quality.

The benefit of including social circles in the coaching process is two-fold. It not only attempts to overcome the problem of the users who are predominantly elderly and are typically averse to using modern technology, but also increases the number of their social interactions, which will ideally also improve their well-being.

The SAAM coaching system covers four coaching domains: *sleep quality*, *mobility*, *everyday activity (cooking, household, and outside)*, and *social activity*. While the SAAM coaching system is quite complex, and describing the details of its working is outside the scope of this paper, it was designed specifically to be easily extended to other application domains.

In the following paragraphs, we only provide a high level description that should allow the reader to understand our preference learning approach and its integration with the coaching system. The basis of the SAAM coaching system are *coaching pipelines* [22,23]. For each of the coaching domains, one or more separate coaching pipelines are implemented, each of which consists of several components, as depicted in Figure 2.

In essence, based on measurements from a set of ambient and wearable sensors (smart electricity meter, inertial measurement unit (IMU) for sleep monitoring, sound detector, wearable IMUs, smartphone, etc.), a set of features is computed, and these features are used as criteria or inputs for the next three components: *situation assessment*, *coaching action modeling*, and *coaching rendering modeling*. Each of these components is realized as a *multi-criteria decision-making (MCDM)* model. As the name implies, the situation assessment model assesses the user’s situation at a given point in time.

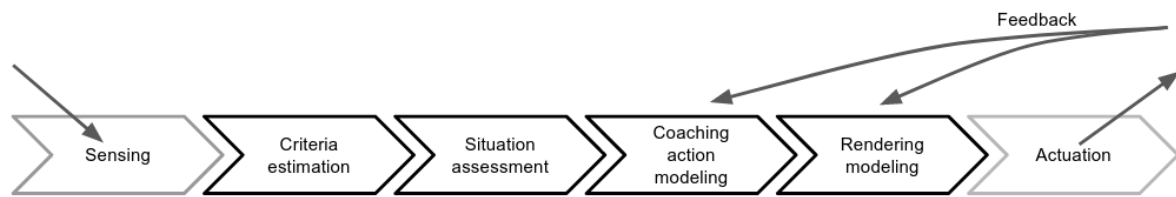


Figure 2. A generalized diagram of coaching pipelines, the basis of the SAAM coaching system.

For example, for sleep quality coaching, sleep is monitored with an IMU attached to bed, and from these measurements criteria such as *sleep latency* and *sleep efficiency* are assessed. These criteria are the inputs to the situation assessment model, and its output is an assessment whether the user slept well during the previous night or not. The coaching action model uses this information, combines it with the previously computed criteria and historical data, and proposes the most suitable coaching action. An example of a coaching action for the sleep quality domain is “*Go to bed only if sleepy*”.

With the coaching action identified, we must also determine its rendering. This is the task of the coaching rendering model. Given a coaching action, this model again uses the computed criteria and historical data, as well as specific user’s preferences as selected by the user in their profile. For example, a user might not want to share information about their sleep quality with their social circles, and decides to receive all sleep quality coaching directly and not through their SUs. In all cases, the coaching rendering model aggregates all this information and outputs a specific rendering for the chosen coaching action.

For example, the coaching action mentioned above, “*Go to bed only if sleepy*”, can be displayed to the PU directly using multiple persuasion strategies. One such strategy is the *suggestive persuasion strategy* which, for example, results in the following message: “*You might sleep better if you only go to bed if you feel sleepy*”.

Finally, the actuation module takes care of presenting the message to the user, for example, by displaying it on the screen of a device or as a voice message played through the ambient sensor. An example of an on-screen display in a web browser is shown in Figure 3. In addition, the actuation module also enables the collection of the user’s feedback regarding each coaching action: a user can *accept*, *decline*, or *cannot perform* the suggested coaching action. As described later in Section 4, this forms the basis of our preference learning approach.

Message	Date	Feedback Buttons
Danes brez svetovanja	20. 11. 2020	
Vaši podatki od tega tedna kažejo, da ste v postelji 22 minut, preden zaspite. Morda bi bilo bolje, če bi šli spat, ko ste zaspani.	20. 11. 2020	Končano, Ne morem, Zavrni
Morda boste spali bolje, če greste v posteljo, ko ste zaspani.	19. 11. 2020	Ne morem

Figure 3. User interface of the SAAM coaching system (in Slovene) with the feedback buttons shown: green for *accept*, red for *decline*, and white for *cannot perform*.

The three MCDM models, i.e., the situation assessment, coaching, and rendering models, are designed in collaboration with domain experts and are implemented as hierarchically structured IF–THEN rules. These rules can produce one or more possible coaching actions (or its renderings), if the experts identified more than one suitable coaching action (rendering) for a given situation. Selection among more than one option can be operated randomly using a suitable probability distribution. By default, this distribution is uniform, however, it can be guided using preference learning, as is described below.

3.2. Preference Learning Targets

Above, we have described that we can adapt the probability distribution of selected coaching (rendering) actions and that by modifying it we can influence specific aspects of coaching. In the setting used during the SAAM project, we can influence three separate coaching aspects, which we call *preference learning targets* (PLTs). These are:

Coaching action. Selecting a coaching action based on user’s preference is only feasible in situations where several coaching actions are valid and appropriate. This requires that the coaching action decision model is probabilistic and a coaching action is selected stochastically according to some probability distribution.

Persuasion strategy. The coaching actions used during the SAAM project were designed according to two persuasion strategies, i.e., *suggestion* and *self monitoring*, under the hypothesis that different persuasion strategies can be more effective for different users. The persuasion strategy is selected using the coaching rendering model.

Interaction target. One of the defining ideas of SAAM is that coaching actions can be delivered through the primary user’s social circle, that is, their assigned secondary users (SU). As people differ, some of the primary users might prefer this approach, while to others might prefer direct delivery of coaching messages (PU). As such, this preference is a natural fit for personalization. In the context of the SAAM system, each user decides whether the coaching messages should be delivered to them directly, through secondary users or both.

The preference learning module described below is designed to specifically address these targets, though others could easily be added. In particular, the probabilities obtained from the preference learning module, can be directly used in the probabilistic selection of a coaching action or rendering for the coaching action PLT or persuasion strategy for its PLT. Finally, for the interaction PLT, if the user has selected *both* delivery options, the final choice can be made using the probability obtained from the preference learning module.

3.3. Integration in the Coaching System

The preference learning module is integrated into the SAAM coaching system in three steps, as follows.

Collection of the learning data. When we first start employing the coaching system, we have no data that would contain information about user’s preferences and that we could use for learning, i.e., we encounter the cold start problem. Therefore, we need to collect such data, and we can achieve this by using the coaching system with any applicable preferences being selected by random. Preferably, this random selection should follow a uniform probability distribution in order to cover the entire space of possible preferences, that is, we want all the possible options to occur so we can better assess the preferences and improve the acceptance of the coaching actions. Such a uniformly distributed learning dataset also eases model learning and typically results in a more accurate model.

Learning of the acceptance model. Once enough learning data has been collected, we can learn a model that predicts user’s acceptance of the received coaching actions in the particular context which they received it in. As we will see later, this model can provide an assessment of the likelihood that each possible coaching action (with a

selected rendering) will be accepted by the user. These probability estimates are then used in the next step.

Using predictions of the acceptance model. The predicted probabilities from the acceptance model can then be used to instantiate the probability distributions for each of the PLTs mentioned above in Section 3.2. Consequently, coaching actions and their renderings should then be adapted to be closer to the user's actual preferences. Particular implementation details will be presented in Section 4.2.

The above three steps can be repeated during the deployment of the coaching system arbitrarily many times and in different settings. In particular, we need to decide whose preferences we want to model, i.e., the data of which users will be used for learning. Generally, these users are likely also the users for which the model will then be used. Notably, however, different settings are also possible, e.g., we might learn a model on a previously selected group of users and then apply the learned preference to a new user that is in some way similar to the original group.

In our case, we can opt for learning the preferences of all users, or preferences of individual users. Given the relatively low amount of real-world data that we have recorded during the SAAM project, as the preference learning module was defined and developed fairly late into the project, the models presented below in Section 4.3 are learned on data from all users, but the procedure is the same in the case of individual user data.

4. Preference Learning

In our preference learning approach, we do not learn the preferences or data instance ranking functions directly as in some other approaches [24], but instead develop a single classification model which models user feedback, which is also valuable for problem understanding and validation, and use its classifications or, more specifically, probability of positive feedback as the inferred preference estimate. After describing the components of the preference learning module, we illustrate the procedure on a sleep quality coaching pipeline and specifically address the three presented PLTs. However, the same procedure is applicable to other pipelines and PLTs in the SAAM coaching system.

4.1. Learning Data

We aim to learn about user's preferences and we can do this based on data comprising examples of different coaching actions issued with different renderings and user feedback reported in each of these cases. Therefore, the data needs to include attributes (or variables) describing: (I) the specific variable for the values of which we are interested in learning the user's preference (PLTs), (II) the context of the coaching action, and (III) the user feedback.

As explained above, PLTs are the coaching action (when multiple are viable), the persuasion strategy and the interaction target. The user feedback is the feedback provided about a specific coaching action that was issued. In the SAAM context, the feedback is provided by the users through a tablet application, where they can select to either *accept*, *decline*, or *cannot perform* for each received coaching action. Finally, the context of the coaching action are all other variables that are available to the preference learning system, such as situation assessment criteria or user profile, which the coaching action may feedback. Furthermore, we can also include variables collected or calculated in pipelines from other domains, if we believe that they might influence user's preferences.

In the context of the SAAM sleep quality coaching pipeline, the collected data consist of sensor measurements, computed features, model outputs, and resulting coaching actions. The data comprise 240 coaching actions addressed to 15 different primary users during a roughly three month period. In addition to the variables describing quality of sleep, we have also included some variables describing cooking activity that were collected in parallel during the operation of one of the activity pipelines, namely the cooking activity pipeline.

Our motive for including cooking-related variables is to demonstrate that additional variables describing the context of a coaching action are easily added even from outside

the particular pipeline, though it is also not unreasonable to expect sleep quality can be linked to cooking habits. All the attributes of our dataset are presented in Table 1. Note that, as we are interested in increasing the number of successful coaching actions, i.e., those that are completed, we consider both *decline* and *cannot* as unwanted results, and merge them into a single value, covering both cases.

Table 1. Attributes of the sleep quality coaching dataset comprised of 324 issued coaching actions, some of the context attributes in the collected data set have missing values.

Attribute Name	Possible Values	Missing	Role
Coaching Action	Get up from bed if awake, Go to bed only if sleepy, See your doctor, Adjust bedroom temperature, Avoid drinking 2 h before sleep	0%	PLT
Persuasion Strategy	Suggestion, Self-monitoring	0%	PLT
Interaction Target	PU, SU	0%	PLT
Predicted Cooking Activity	None, Once, Twice or more	91%	Context
Recorded Cooking Activity	None, Once, Twice or more	91%	Context
Cooking Activity Situation	Usual cooking, Less cooking	91%	Context
Subjective Sleep Quality	Numeric (0–3)	8 %	Context
Sleep Disturbance Type	None, Medical, Bathroom, Temperature	8%	Context
Napping During Day	Yes, No	8%	Context
Sleep Efficiency	Numeric	8%	Context
Sleep Latency	Numeric	8%	Context
User Feedback	Accept, Cannot or decline	0%	Target

4.2. The Learning Approach

Once the learning data is available for preference learning, we can start by learning a classification model. Our goal is to learn a predictive model that is based on the input variables (PLTs and context) and predicts the user feedback, i.e., whether the user will accept (and hopefully fulfil) the issued coaching action, or not. Therefore, we are learning a predictive model M for the variable *user feedback*:

$$\text{User Feedback} = M(\text{PLTs, Context}).$$

In principle, we can use an arbitrary multi-variate learning or modelling approach, however, we prefer approaches that, in addition to the predicted value, also provide a probability estimate for each of the possible outcomes. Such probability estimates make it considerably easier to construct the probability distributions that can be used in coaching activity and coaching rendering models.

A predictive model M can only be used to predict whether the user is likely to fulfil a specific received coaching action or not, given the values of the PLTs and the context variables. However, what we require is an estimate of user's preferences for coaching. This is achieved using the following procedure. When a coaching action is being issued (or selected), values of all context variables are available, as this is how the coaching system is designed. For each possible value combination of the applicable PLTs, we prepare a single data instance using the values of the context and the selected values of the PLTs. This results in a set of data instances with identical values of context variables, where each data instance corresponds uniquely to a selection of PLTs, which are exhaustively covered by the generated instances.

Following the construction of the instances, we run all the instances through the predictive model, i.e., we provide them as input to the model, and collect predictions (and their probabilities) for the user feedback (coaching action acceptance). Based on these probabilities, we easily construct probability distributions required in coaching activity and coaching rendering models (Section 3.3).

Let us illustrate this procedure on an example. For simplicity's sake, we will only address two PLTs, *persuasion strategy* with possible values *suggestion* and *self-monitoring*, and *interaction target* with possible values *PU* and *SU*. For this situation, we construct four input instances:

$$\begin{aligned} I_1 &= (\text{Per. Strategy} = \text{Suggestion}, \text{Int. Target} = \text{PU}, \text{Context}), \\ I_2 &= (\text{Per. Strategy} = \text{Suggestion}, \text{Int. Target} = \text{SU}, \text{Context}), \\ I_3 &= (\text{Per. Strategy} = \text{Self-monitoring}, \text{Int. Target} = \text{PU}, \text{Context}), \\ I_4 &= (\text{Per. Strategy} = \text{Self-monitoring}, \text{Int. Target} = \text{SU}, \text{Context}). \end{aligned}$$

Running these four instances through a (hypothetical) predictive model M gives us four predictions for user feedback (acceptance) and probability estimates for user's acceptance (P_{acc}):

$$\begin{aligned} M(I_1) &= \text{Accept}; P_{\text{acc}}(I_1) = 0.75, \\ M(I_2) &= \text{Accept}; P_{\text{acc}}(I_2) = 0.65, \\ M(I_3) &= \text{Decline}; P_{\text{acc}}(I_3) = 0.25, \\ M(I_4) &= \text{Decline}; P_{\text{acc}}(I_4) = 0.15. \end{aligned}$$

Note that probability estimates obtained by most machine learning methods are not true probabilities and are not normalized; however, this is not a problem for our intended use. We transform the above values into distributions for both PLTs in a simple manner, by aggregating the probabilities for all possible PLT values:

$$\begin{aligned} D(\text{Per. Strategy}) &= \left(\frac{P_{\text{acc}}(I_1) + P_{\text{acc}}(I_2)}{\sum_{i=1}^4 P_{\text{acc}}(I_i)}, \frac{P_{\text{acc}}(I_3) + P_{\text{acc}}(I_4)}{\sum_{i=1}^4 P_{\text{acc}}(I_i)} \right) \\ &= (0.78, 0.22), \\ D(\text{Int. Target}) &= \left(\frac{P_{\text{acc}}(I_1) + P_{\text{acc}}(I_3)}{\sum_{i=1}^4 P_{\text{acc}}(I_i)}, \frac{P_{\text{acc}}(I_2) + P_{\text{acc}}(I_4)}{\sum_{i=1}^4 P_{\text{acc}}(I_i)} \right) \\ &= (0.56, 0.44) \end{aligned}$$

These two distributions can now be used directly in the coaching rendering model. In this example, the adapted coaching rendering model will now significantly favor the suggestive persuasive strategy and will slightly prefer coaching to the PU.

4.3. Preference Model for Sleep Quality Coaching

As mentioned above, this approach works with any statistical or machine learning method that learns a model that outputs probabilities in addition to classifications. Many approaches such as neural networks, Bayes-based classifiers, or kernel-based learning methods could be used, however in this example we have specifically chosen decision trees and a random forest of decision trees. This is based on the fact they possess certain features that benefit the example in descriptiveness as well as provide a deeper insight into which of the features are key to determining the coaching outcome. In particular, both methods produce multivariate predictive models, however decision trees learn a model that is particularly interpretable, i.e., we can manually and even visually inspect the model to understand how it calculates predictions. Conversely, a random forest does not provide learn such a clean model, yet it provides us with an easily obtainable ranking of the input variables. With the ranking we can, in theory, understand how large an effect each variable, in particular, each PLT value, has on whether or not the coaching action will be accepted. All the presented models below were learned using the Weka machine learning toolbox [25].

We start by learning a decision tree model using the J48 algorithm [26]. By selecting default parameters values, the resulting model was a somewhat large tree (29 nodes and 18 leaves) than could not be presented here. To address this, we pruned it slightly, which affects lower-laying nodes, while the tree does not change significantly in the vicinity of the root. Decision tree pruning is a standard procedure that removes the less informative attributes from the model in order to reduce complexity of a model, and usually results in a more robust model that generalizes the learning data better. Notably, the attributes appearing in the root and nearby are the ones that most influence the final prediction, so pruning the tree usually does not incur a significant change in the resulting probabilities and consequently the preference function estimation.

The pruned tree is presented in Figure 4. Judging by the model, the variables that most influence user acceptance are persuasion strategy used for the rendering of the coaching action, the user's sleep efficiency, the type of their sleep disturbance, their subjective sleep quality and whether they napped during the day or not. The leaves of the tree (colored in orange) contain the predicted value, the number of examples that reached that leaf during learning and the number of examples that were not properly classified by this leaf. Notably, the numbers are not integers due to missing values in the learning data. When the value of an attribute appearing in the split test of the tree is missing, the example is split to both branches.

Furthermore, these values can be used to assess the probability of a classification of a new example: we note which leaf was used for classification of a given example and assume that a leaf with less misclassifications provides predictions with higher reliability. Note that only one PLT (persuasion strategy) appears in the model, and thus this model cannot be used to adapt probability distributions of the other two PLTs (coaching action and interaction target). Other types of machine learning models can provide such probabilities for all attributes; they are, however, much less illustrative.

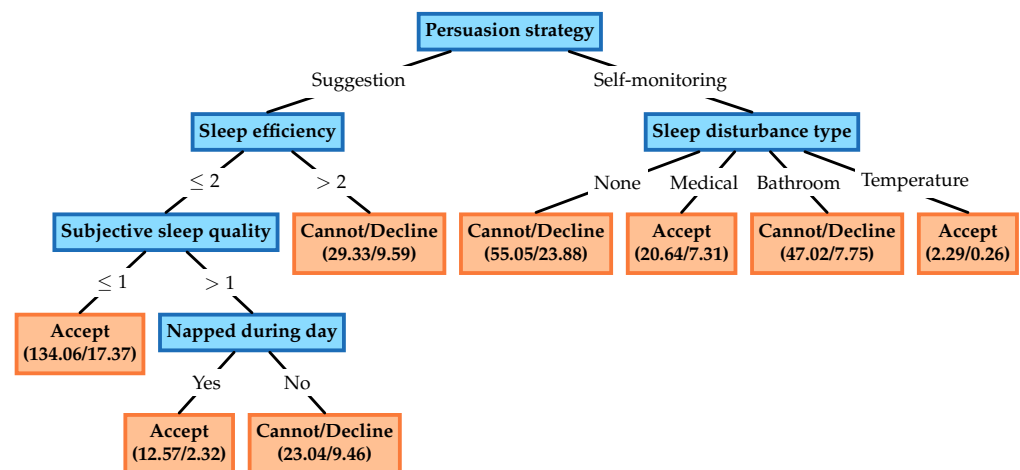


Figure 4. Decision tree for predicting user acceptance of a sleep quality coaching action.

The second learning algorithm that we have applied to the same data is the random forest method. This is a well-known ensemble method [27] that is also decision tree-based, but as an ensemble method it combines multiple trees together and, as such, the final model is no longer interpretable. However, using this method, we can generate feature ranking scores for input variables and use them to estimate the influence that each of the variables has on the prediction. In our case, the algorithm produces the scores presented in Table 2, where higher values imply a larger effect on the final prediction. The resulting ranking confirms the importance of the variables identified by the decision tree (persuasion strategy, sleep efficiency, type of sleep disturbance, subjective sleep quality, and napping during the day), although they suggest that interaction target is also important.

Table 2. Ranking of the input variables with the random forest method for user acceptance of the sleep quality coaching actions. PLTs are typeset in bold.

Ranking Score	Input Variable
0.19	Subjective Sleep Quality
0.17	Persuasion Strategy
0.17	Napping During Day
0.16	Interaction Target
0.16	Sleep Latency
0.16	Sleep Disturbance Type
0.14	Sleep Efficiency
0.08	Coaching Action
0.04	Recorded Cooking Activity
0.03	Predicted Cooking Activity
0.02	Cooking Activity Situation

Focusing on the PLTs, we observe that the likelihood of the acceptance of the coaching action depends mostly on the persuasion strategy and the interaction target and significantly less on the coaching action itself. Of course, all these results need to be further critically considered and are not to be used to draw any final conclusions. In particular, this is due to the fact that they were obtained from a small data set, with some missing values, and the variable distributions, notably of some PLTs, are quite skewed.

Another point to discuss is the accuracy of these models. Using the 10-fold cross-validation procedure, the presented decision tree model has a 71.6% accuracy (or 0.77 AUC; here the value above 0.5 means that the model is better than chance) and the random forest model has a 77.8% accuracy (0.84 AUC). These values are not stellar, but, in addition to the already mentioned small dataset, we need to be aware of two things. First, the problem that we are modelling is hard and the data that we can collect about this problem likely do not contain all the information that we would need to reliably predict the preferences. Second, we do not use these predictions to construct the coaching models from scratch, or to overrule the essential parts of the expert-based models, but we use them to improve the acceptance of the coaching actions within the specified flexibility. In this regard, if these models are better than chance, and we have shown that they are, they are already useful and can improve the results of coaching.

5. Discussion

With a novel approach such as this, however, many of its aspects warrant some discussion. We identify three particular areas that should be taken into consideration when using a system like this. In particular, we examine the potential automation of such a system, its broader use in general (non-SAAM) coaching systems and how the efficacy of such a system could be properly evaluated.

There are several ways in which the learned preferences can be used to adapt the coaching system components and tailor them to the users. A procedure of this kind could easily be integrated into the system in a way to allow completely automated adaptation. Although alluring, automation of such a procedure that changes the reasoning components of the system can potentially be harmful, as it can cause the system to gradually change into a state in which it operates too differently to how it was envisioned. We could make the adaptation approach cautious, e.g., not to be able to change zero probability coaching actions to non-zero ones, or not allowing preference probabilities to fall under a certain threshold. However, as coaching affects people and as the coaching systems and coaching feedback are not yet fully understood phenomena, we believe that, regardless of the execution mode, manual or automated, the adaptations to user preferences should be monitored by the system operators and domain experts and not left to be changed according to data without human oversight.

While we designed the preference learning module specifically with the SAAM coaching system in mind, the approach could be applied also to other similar systems. As mentioned above, the main innovation of the SAAM project was the inclusion of the primary user's social circles in the delivery of the coaching actions. Other components, such as the situation assessment and coaching action selection model (whether MCDM or otherwise), are fairly standard in other coaching systems. As these are sources of data about context and even directly of the coaching action, i.e., one of the PLTs, the kind of data that we gathered for the SAAM coaching system and applied our preference learning approach to should be feasible to obtain in other systems as well. While the same PLTs might not be applicable directly, any kind of parametrization in the coaching process can be encoded as new PLTs.

A question that arises from the example above is how the efficacy of such a preference learning module could be assessed. Notably, as the system is meant to operate continuously, an evaluation of the module can not be a single time assessment. How we approach evaluation also depends in a major way on how we collect our data, in particular, whether we are joining together data from multiple users or only using data from a single user. Not only will the later result in a considerably slower accumulation of data, it also makes any kind of testing against a control "group" more cumbersome.

In a long running system, we can then suggest the following methodology. First, fix an amount of time, an *evaluation window*, at which the efficacy of the preference learning module will be periodically assessed and the system's preference updated. When a new evaluation window starts, some measurements should be made using a control group preference, while others should be made with the latest inferred preferences. One approach, in the multiple-user data setting, would be to randomly select some users and use control group preferences, whether they be the default (uniformly random) or perhaps preferences from the previous evaluation windows. However, using this kind of control in a system that is meant to be actively beneficial to the users would leave some users with a worse experience, as not all possible effort would be made to actually achieve the execution of the coachings. This approach also cannot cover the single-user data setting.

As we still want to compare the new preferences to some baseline or previously learned preferences, we suggest the following. Instead of randomly selecting users which receive coachings using the baseline/comparison preferences, we randomly select individual coaching actions this way. The probability of selecting the use of baseline/comparison preferences should be relatively low, to not impact the user's interaction with the system to a large extent. This probability should be chosen in tandem with the evaluation window, to ensure that the baseline/comparison dataset is large enough for meaningful comparisons, while still small enough that the impact of the concurrent evaluation procedure does not impact the users in a significant way. Notably, this methodology also conveniently covers the case of single-user data, as it can be applied to that case in the same way.

6. Conclusions

This paper describes the processes for assessment and integration of user preferences in a coaching system. It outlines the aspects of the coaching system that we can adapt according to user preferences, presents the machine learning-based approach to preference learning, and demonstrates it on a real-world example. While the presented results are illustrative, they confirm the feasibility and usefulness of the proposed approach. Finally, we provide some deliberations on the automation, generalizability, and evaluation of such an approach.

It is exactly in these areas that we recognize possibilities for further study. Applying our approach to more data, whether with the SAAM coaching system or another, evaluating how it affects the successful execution of coaching actions over a prolonged period of time, while also seeking to identify which components can be made automated and to what extent.

Namely, the limitations of the proposed approach are not only technical, such as the data requirements and the cold start problem issues as explained in Section 3, but are also to some extent bound by the aspects of coaching goals and ethical considerations. For example, while the changes to persuasion strategies or recipients of the coaching actions are relatively benign and only increase or decrease the effectiveness of coaching, the coaching actions themselves could be much more sensitive (not all equally so, of course). There are users who, upon receiving appropriate coaching to correct a specific issue, decline to follow the suggestions, as this would mean abandoning much liked (although potentially detrimental) routines. Of course, it is up to the users to prioritize their goals and wishes and the system should adapt to the users, but it is difficult to prevent reaching a point where the system is coaching more what the users would like to be coached than what would help them to reach their goals. This calls for further interdisciplinary research on coaching, but automated data-based adaptation to preferences also calls for active involvement and interaction of the involved humans (PUs, SUs, and domain experts) in this aspect of the coaching system, at least to some extent.

Author Contributions: Conceptualization, methodology, and investigation, M.Ž. and B.Ž.; software, A.O.; data curation, P.R.; writing—original draft preparation, M.Ž. and B.Ž.; writing—review and editing, All authors; visualization, A.O.; and funding acquisition, M.Ž. and B.Ž. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the EU via the H2020 project SAAM (Grant No. 769661) and by the Slovenian Research Agency through research core funding of the program P2-0103.

Institutional Review Board Statement: All subjects gave their informed consent for inclusion before they participated in the study. The study was following the guidelines of the Declaration of Helsinki as detailed in https://www.saam2020.eu/documents/Deliverables/User_Interaction_Ethical_Review_Report.pdf (accessed on 14 January 2021). The protocol of the study was approved by the competent Ethics Committees of the piloting partners of the SAAM Project (EU Horizon 2020 programme, grant agreement No. 769661). The adopted ethical procedures and approvals used in the project are available at https://www.saam2020.eu/documents/Deliverables/Ethics_codex.pdf (accessed on 14 January 2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data used for generating the example models presented in the paper is available at <https://zenodo.org/record/5807971#.YgC4IfgRU2w> (accessed on 31 January 2021) under the CC BY 4.0 License.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Žnidaršič, M.; Osojnik, A.; Rupnik, P.; Ženko, B. Improving Effectiveness of a Coaching System Through Preference Learning. In Proceedings of the 14th Pervasive Technologies Related to Assistive Environments Conference, Corfu, Greece, 29 June–2 July 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 459–465. [CrossRef]
2. Luce, R.D.; Raiffa, H. *Games and Decisions: Introduction and Critical Survey*; Publications of the Bureau of Applied Social Research of the Columbia University; Wiley: New York, NY, USA, 1957.
3. Fürnkranz, J.; Hüllermeier, E. (Eds.) *Preference Learning*; Springer: Berlin, Germany, 2010. Available online: <https://link.springer.com/book/10.1007/978-3-642-14125-6> (accessed on 31 January 2021).
4. Fürnkranz, J.; Hüllermeier, E.; Rudin, C.; Slowinski, R.; Sanner, S. *Preference Learning (Dagstuhl Seminar 14101)*; Dagstuhl Reports; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Wadern, Germany, 2014; Volume 4.
5. Greco, S.; Figueira, J.; Ehrgott, M. *Multiple Criteria Decision Analysis*, 2nd ed.; Springer: New York, NY, USA, 2016.
6. Keeney, R.L.; Raiffa, H.; Meyer, R.F. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*; Cambridge University Press: Cambridge, MA, USA, 1993.
7. Farquhar, P.H.; Keller, R.L. Preference intensity measurement. *Ann. Oper. Res.* **1989**, *19*, 205–217. [CrossRef]
8. Braziunas, D.; Boutilier, C. Elicitation of factored utilities. *AI Mag.* **2008**, *29*, 79. [CrossRef]
9. Salo, A.A.; Hamalainen, R.P. Preference ratios in multiattribute evaluation (PRIME)-elicitation and decision procedures under incomplete information. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2001**, *31*, 533–545. [CrossRef]

10. Siskos, Y.; Grigoroudis, E.; Matsatsinis, N.F. UTA methods. In *Multiple Criteria Decision Analysis*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 315–362.
11. Bous, G.; Pirlot, M. Learning multicriteria utility functions with random utility models. In *International Conference on Algorithmic Decision Theory*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 101–115.
12. Yang, J.B.; Sen, P. Preference modelling by estimating local utility functions for multiobjective optimization. *Eur. J. Oper. Res.* **1996**, *95*, 115–138. [[CrossRef](#)]
13. Siskos, Y.; Spyridakos, A.; Yannacopoulos, D. Using artificial intelligence and visual techniques into preference disaggregation analysis: The MIIDAS system. *Eur. J. Oper. Res.* **1999**, *113*, 281–299. [[CrossRef](#)]
14. Branke, J.; Greco, S.; Słowiński, R.; Zielniewicz, P. Learning value functions in interactive evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **2014**, *19*, 88–102. [[CrossRef](#)]
15. Cruz-Reyes, L.; Fernandez, E.; Rangel-Valdez, N. A metaheuristic optimization-based indirect elicitation of preference parameters for solving many-objective problems. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 56–77. [[CrossRef](#)]
16. Fürnkranz, J.; Hüllermeier, E. Preference Learning and Ranking by Pairwise Comparison. In *Preference Learning*; Fürnkranz, J., Hüllermeier, E., Eds.; Springer: Berlin, Germany, 2010; pp. 65–82.
17. Jannach, D.; Zanker, M.; Felfernig, A.; Friedrich, G. *Recommender Systems: An Introduction*; Cambridge University Press: Cambridge, MA, USA, 2010.
18. Baier, D. Bayesian Methods for Conjoint Analysis-Based Predictions: Do We Still Need Latent Classes? In *German-Japanese Interchange of Data Analysis Results*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 103–113.
19. Todd, D.S.; Sen, P. Directed multiple objective search of design spaces using genetic algorithms and neural networks. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, Orlando, FL, USA, 13–17 July 1999; Morgan Kaufmann: San Francisco, CA, USA, 1999; Volume 2, pp. 1738–1743.
20. Misitano, G. Interactively Learning the Preferences of a Decision Maker in Multi-objective Optimization Utilizing Belief-rules. In *Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Canberra, Australia, 1–4 December 2020; IEEE: New York, NY, USA, 2020; pp. 133–140.
21. Lete, N.; Beristain, A.; García-Alonso, A. Survey on virtual coaching for older adults. *Health Inform. J.* **2020**, *26*, 3231–3249. [[CrossRef](#)] [[PubMed](#)]
22. Dimitrov, Y.; Gospodinova, Z.; Wheeler, R.; Žnidaršič, M.; Ženko, B.; Veleva, V.; Miteva, N. Social Activity Modelling and Multimodal Coaching for Active Aging. In *Proceedings of the 12th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, Rhodes, Greece, 5–7 June 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 608–615.
23. Žnidaršič, M.; Ženko, B.; Osojnik, A.; Bohanec, M.; Panov, P.; Burger, H.; Matjačić, Z.; Debeljak, M. Multi-criteria Modelling Approach for Ambient Assisted Coaching of Senior Adults. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Vienna, Austria, 17–19 September 2019; INSTICC, SciTePress: Vienna, Austria, 2019; Volume 2, pp. 87–93.
24. Hüllermeier, E.; Fürnkranz, J. Preference learning and ranking. *Mach. Learn.* **2013**, *93*, 185–189. [[CrossRef](#)]
25. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; The Morgan Kaufmann Series in Data Management Systems; Elsevier Science: Amsterdam, The Netherlands, 2016.
26. Quinlan, R.J. *C4.5: Programs for Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993.
27. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]